

Grails

The Grails logo, featuring a stylized golden chalice or cup with several small golden spheres floating around it, positioned in the top right corner of the cover.

技术精解 与 Web 开发实践

宋友 梁士兴 黄璐 王鑫 等编著



清华大学出版社



Grails



技术精解 与 Web 开发实践

宋友 梁士兴 黄璐 王鑫 等编著



清华大学出版社
北京

内 容 简 介

Grails 是继 J2EE、PHP、ROR 等技术之后又一个成功的 Web 框架。本书系统讲解了使用 Grails 技术快速进行 Web 开发的知识。本书内容分为四大部分, 第一部分介绍了 Grails 必备的基础知识, 包括环境配置、Groovy 语言基础、HelloWorld 程序开发实例等; 第二部分以迭代的方式, 设计并逐步完善了一个购物车应用, 介绍了 Grails 基础知识; 第三部分对 Grails 各个部分的细节进行了深入讨论; 第四部分对 Grails 的实现原理进行了剖析, 通过分析 Grails 的源码, 帮助读者体会出 Grails 如此神奇的奥妙所在。

本书可作为大学本专科软件工程等专业教材, 其适用的课程可以为 Web 开发与实践、软件工程实践等。本书也可供 Web 开发与应用的工程技术人员和爱好者参考。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目 (CIP) 数据

Grails 技术精解与 Web 开发实践 / 宋友等编著. —北京: 清华大学出版社, 2009.9
ISBN 978-7-302-20187-8

I. G… II. 宋… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2009) 第 076421 号

责任编辑: 夏兆彦

责任校对: 徐俊伟

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京市清华园胶印厂

装 订 者: 三河市兴旺装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 16.5 字 数: 408 千字

版 次: 2009 年 9 月第 1 版 印 次: 2009 年 9 月第 1 次印刷

印 数: 1~4000

定 价: 33.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系
调换。联系电话: (010)62770177 转 3103 产品编号: 031598-01

FOREWORD

推荐序

互联网的发展带来了新的科技革命，Web 2.0 的概念更是将网络技术和应用推向了新的高潮。在这个技术日新月异发展的年代，各种框架、各种概念、各种思想呈现出了百花齐放局面，虽然无比繁荣，却让人十分眼花缭乱。选择一种最适用的 Web 框架，可以给开发时间、开发成本、开发质量带来巨大的影响。

当今世界是一个创造奇迹的时代。J2EE 创造了一个奇迹，使得开发企业级的 Web 应用变得快速、可靠、经济，从而成为近年来的主流 Web 开发框架。RoR 的诞生创造了另一个奇迹，它的开发效率是 J2EE 的 5 倍以上，而且相比 PHP 技术，它有着更好的代码组织结构，更容易写出高质量的代码。

Grails 是继 RoR 之后的又一个奇迹：它不但具备 J2EE 和 RoR 的诸多优点，还解决了 RoR 不能有效适用于广大 Java 用户的一大难题。目前，Grails 已得到了业内广泛地认可。在 2008 年 11 月 11 日，以 Spring 框架而闻名于天下的 SpringSource 宣布收购主导 Grails 开发的 G2one 公司，并承诺在收购之后，会投入更大的力量去改进 Grails 并且对 Grails 的全球用户提供商业支持。正如著名的 ququjoy 的评论所述：“SpringSource 将 G2One 收购之后，以 Spring 为底层框架的 Grails 必将迎来自己的春天”。对于还在观望的用户，还犹豫什么呢？赶快加入到 Grails 的使用者队伍中吧，充分享受 Grails 带来的实用与便捷。

目前，市面上关于 J2EE 的书籍数不胜数。关于 RoR 的书籍也至少有数十本。然而，也许很快将引领潮流的 Grails 实在太新了，它的资料主要还在 Grails 官方网站上能获得。关于 Grails 的中文书籍仅有两本外版图书，并且介绍的是低版本的 Grails。这对于广大爱好 Web 开发、并渴望学习 Grails 这门新技术的读者来说，不能不说是一个遗憾。

可喜的是，这本书问世了，该书的作者们自己使用 Grails 开发了许多成功的应用。为了撰写该书，他们亲自编写了一个典型的网上购物车的 Web 应用实例。通过 Web 实例制作的示范和讲解，由浅入深地分析了 Grails 的神奇，实践性非常强。同时，本书也对 Grails 的源代码、基于 MVC 的 Web 开发原理进行了深入的分析，亦不失理论深度。该书的面市相信会为 Web 开发爱好者带来一个小的惊喜。

该书优美地结合了理论与实践。不失为一本优秀的教科书和程序员与项目经理的参考手册。值得一读。

孙 伟

北京航空航天大学软件学院院长

2009 年 4 月 5 日

FOREWORD

前言

目前,虽然 J2EE 仍然是企业级 Web 开发的主流,但越来越多的企业和个人在开发新项目时更倾向于尝试使用 RoR,因为用 RoR 开发 Web 应用,可以更快捷、更简单。仿佛在一夜之间, RoR 成为了 Web 框架领域内最耀眼的明星, RoR 的一些理念,如约定优于配置,已经成为了新框架设计的金科玉律。

然而, RoR 对于世界上数量最大的 Java 用户来说,并不足够友好。第一, Ruby 语法与 Java 类语言有较大区别,需要克服较大的障碍才能掌握。第二,历史遗留项目问题, Java 语言的广泛流行使得大量既有项目是用 Java 开发的,而且这些项目还需要被继续支持和扩展,扩展历史遗留项目的任务, RoR 显得无能为力。第三,性能问题, Ruby 语言相对于 Java,其执行性能较差。上面这些因素,都会制约 RoR 在实际项目中的使用。正是在这一背景下, Grails 诞生了。

Grails 的全称为 Groovy on Rails,是一种基于 Groovy 语言的 Rails 类框架。它继承了 RoR 的绝大多数优点,几乎和 RoR 一样简单。而对于 RoR 的不足,它有着明显的优势。第一, Groovy 语言运行于 JVM,可以透明地执行 Java 程序。Java 程序员可以迅速地掌握 Groovy,同时对基于 Java 的历史遗留项目,可以得到非常好的支持。第二, Groovy 语言的执行性能虽然不及 Java,仍然比 Ruby 有较大的优势。另一方面, Grails 虽然名为 Groovy on Rails,但实际上 Grails 的主要开发语言是 Java, Grails 框架有的绝大多数代码是用 Java 编写的,因而, Grails 的性能就更可以得到保障。

Grails 技术凭借它卓越的设计,给用户提供了一种高效、简单易用、一栈式 Web 开发的框架。利用 Grails 技术,可以以极低的学习代价,换来很高的开发效率。使用 Grails 开发 Web 应用,需要使用的编程语言是 Groovy 语言。Groovy 是运行于 JVM 之上的程序设计语言,近些年发展迅猛。对于 Java 用户,要想掌握 Groovy,是非常容易的,学习曲线也非常平缓,甚至不需要学习,仅凭借 Java 的基本功,就可以把 Grails 驾驭得非常好。因此,极力推荐正在使用 Java 开发 Web 应用的公司、学生、个人用户学习并使用 Grails。对于没有掌握 Java 技术,但很想学习 Web 开发技术的用户,那不妨就直接从 Grails 开始吧, Grails 非常容易掌握。使用 Grails,开发人员就可以更多地把精力花在业务上,而不是技术细节上。

一、阅读本书需要的预备知识

现代的 Web 应用通常会涉及到页面设计、数据存储等多个方面。Grails 将这方方面面的问题都考虑了进来，因而说 Grails 是一个一栈式的框架。本书预期的读者，应预备以下几方面的基础知识。

IV

1. HTML 的基础知识。是的，无论如何，HTML 都是 Web 技术的基础，只有具备一定的 HTML 基础，才能设计并输出 Web 页面的基本元素。同时，应该对表单的提交（GET/POST）有所了解。当然，这部分内容非常容易掌握，通过百度 Google，可以轻松地找到许多有用的教程。

2. 数据库的基础知识。Grails 对数据库的操作进行了封装和简化，但开发人员仍然有必要掌握一定的数据库知识，对数据库表、主键、外键、索引，有基本的了解。能够读写简单的 SQL 语句。只有这样，才能真切地体会到 Grails 与生俱来的快捷与便利，而且遇到问题才不会茫然。

3. Java 语言或 Groovy 语言，或者其他类似的面向对象程序语言。使用 Grails 进行开发，需要使用 Groovy 语言。掌握 Java，就意味着在相当程度上掌握了 Groovy。它们二者间比较重要的差异，本书中会进行必要的介绍。当然，对于不了解 Java 但至少掌握了一种编程语言的读者，学习 Groovy 也不会有任何困难，毕竟 Groovy 的设计思想就是设计一种比 Java 更简单易用的编程语言。

4. Hibernate 基础知识（非必需）。本书会对 Grails 的部分实现原理进行深入和分析，会涉及到许多 Hibernate 中的概念。读者如果只是希望用 Grails 开发自己的 Web 应用，则无需要求这一点。

二、关于本书的内容说明

本书的内容包括四大部分。从第一部分到第四部分，按知识的难度和深度，存在一定的递进关系。但各部分也相对独立。

第一部分是入门篇，介绍了 Grails 必备的基础知识，包括：环境配置、Groovy 语言基础、HelloWorld 程序开发实例，以及一些基本概念。对于熟悉 Java 编程的读者，可以快速浏览本部分。

第二部分共 7 章，以迭代增量的方式，完整地设计并开发了一个 Web 应用——网络购物车。该部分内容从最简单的环境准备开始，接下来包括数据创建、数据查询、数据维护与管理、系统后台管理、系统测试、系统部署等。该部分是一个逐步完善的过程，每开始新的一章，就是增加了一些新的功能和知识点。

学习完前两部分，可以掌握 Grails 主要的基础知识，并能开发一些简单的 Web 应用。这可以称得上是 Grails 开发的初级用户了。

第三部分共 6 章，对 Grails 的关键问题进行了比较深入的讨论，对相关的原理进行了介绍。该部分内容包括 Grails 的 MVC 原理、GSP 输出、Web Service 实现、Grails 插件使用等内容。本部分的内容有一定难度和深度，但这些知识也是 Web 开发的基本要素，要开

发真正实用的 Web 应用，掌握这些知识是必要的。

掌握了前三部分内容的读者，可以称得上是中级用户了，能够比较熟练地使用 Grails 开发 Web 应用。

第四部分是最后 5 章，其内容包括：介绍 Groovy 的高级特性、分析 Grails 的源码、讲解 Grails 的插件开发、阐述 Grails 的新特性等。该部分内容难度和深度都较大，是对 Grails 的一些原理进行深入分析。掌握该部分内容，有助于帮助读者体会出 Grails 之所以如此神奇的奥妙所在，在实践中能够通过灵活运用 Grails，并可能修改源码，进行更丰富的二次开发。

在实例分析和原理解析上，本书两者并重，它们分别占了约一半的篇幅。但两者并不是截然分离的，有的时候两者是有机地结合在一起。读者通本书实例的学习实践，可以依葫芦画瓢地开发自己的 Web 应用；通过品味 Grails 的原理，将能开发出更多、更复杂的 Web 应用。读完这本书，读者不仅仅能开发一个网上购物车，同样能开发自己的工作流、表单系统、等等。

由于 Grails 的知识很新，可参考的文献非常少，本书的知识大部分是作者从 Grails 的官方资料中获得的。本书是作者通过阅读和分析 Grails 的源码，亲自开发 Web 应用，编写了较多的实际代码，然后再总结自己从事 Web 开发的一些经验而完成的。本书原创内容较多，因而疏漏之处可能不少，诚恳地请读者批评指正。

三、代码的阅读说明

本书的代码可从网络下载，网址为 www.tup.tsinghua.edu.cn，包括了网络购物车的完整代码。

在本书中，因为购物车的开发是增量进行的，所以随着每一章的代码逐步添加，最后才形成了完整的应用代码。但读者在阅读过程中，阅读并实践到某些章节时，因为还不涉及到后面章节的内容，所以将会发现自己按书上步骤介绍而写出来的代码跟我们提供代码的内容并不一致。

但我们不打算把每一章的代码都单独放在一个文件夹里，写成一个独立的应用。一方面，这样会破坏该应用的完整性；另一方面，也是更重要的，读者在阅读每一章时，可能会不假思索地把我们的代码复制到开发环境里，顺利地让程序运行起来，自己却可能并不明白代码说的是什么。

读者在阅读并实践本书的应用实例时，我们鼓励读者按书上描述的步骤，尽量自己亲自输入相关的代码，并运行程序分析结果。如果有些例子的代码的确非常多，可以从我们提供的代码里摘录部分核心内容。

另外，在分析实例时，有些知识点的代码也比较长，我们在书里也没有全部把代码列出来，读者可以参阅网络文件里的相关内容。

四、写作小札

在 2008 年 4 月初时，我们刚使用 Grails 开发了一个成功的商业应用。在开发该应用的过程中，我们深刻感受到 Grails 中文资料在国内的匮乏，而且原创的 Grails 中文书籍在

国内还没有。另外，市面关于 Grails 的书籍中，基本上都是以 Grails 0.3 为基础来介绍的，但当时主流应用的是 Grails 1.03，两者的差距较大。这给广大 Web 爱好者学习和应用 Grails 带来了困难。于是，我们就萌发了写作此书的想法。

2008 年五一后，该写作计划开始启动，并且该计划得到了清华大学出版社和北航软件学院的大力支持。

我们原计划用大约半年时间完成该书的写作，但事实上花了近一年时间才完稿。实际写作过程中，开发一个完整的 Web 应用实例并不是难事。但在分析 Grails 的原理时，要把 Grails 之所以神奇的本质阐述清楚（例如 Grails 如何通过 Hibernate 进行数据库操作、如何通过 Spring 引入外部的组件、等等），则并不容易，主要是这方面的参考资料非常少，作者不得不花大量的功夫来仔细研读并分析 Grails 的源代码，然后用通俗的语言，并设计一些例程进行解释。

本书的大部分内容是在 2008 年完成的，这其中经历国人大悲的 5.12 地震和大喜的北京奥运会，因此，我们的写作心情也受到巨大的影响，前半段时间是在心情很不平静的情况下完成写作的。整个暑假，我们每天都在坚持写作，当然，期间有伴随着边看奥运会边写作的一段快乐日子。在此期间，我们也承接了一些用 Grails 开发的 Web 应用小项目，当然，那时我们使用 Grails 开发的速度已非常快，效益比率也非常可观。开发这些应用，对于我们进一步理解 Grails，从而更好地写作该书提供了许多有意义的启发。

我们写作的基础曾是 Grails 1.03，但 Grail 的发展很快，寒假前夕，我们的书稿快完时，Grails 1.04 正式版出来了，我们不得不把整本书的相关内容都换成 1.04 版的内容。当时，1.1 版的测试版也有了，于是我们还特别地介绍了部分新版本的新特性。但很遗憾，刚完稿没多久，1.1 的正式版又出来了。我们深刻理解，计算机领域，技术的更新速度永远都比知识传播得要快，因此我们知道，我们难以保证今天出的书里面的内容一定反映今天最新的事情，虽然我们有时可以预测今天或明天将要发生的事情。该书目前以 Grails 1.04 为基础进行介绍，这对于读者学习和使用 1.1 版而言，应该说没有影响。不过我们希望，每隔一段时间后，能不断有新结合 Grails 新版本、反映 Grails 新特性的书籍出现。下一个作者，也许就是你——今天的读者。

该书写完后，本书的作者之一，梁士兴也顺利硕士毕业了。在进行该书的写作过程中，他还完成了自己的学位论文写作、论文答辩、就业等事情，并且，写作这本书也有力地促进了他在学业和就业上的进步。最终，他以优良的毕业成绩通过答辩，并得到了自己满意的工作——获得了 IBM 的 offer。于是，他得到一个感悟：读一本好书、学习一门技术、开发一个项目、撰写一本书和文章，其收获可能是自己也无法预知的一个惊喜。

五、致谢

在本书近一年的写作过程，得到了广大老师、朋友的热心支持。

感谢北京航空航天大学软件学院的领导和老师对该书撰写工作的支持，并提供了良好的写作条件，该项目得到了北京市教育委员会第二类特色专业建设项目支持。

感谢曹媛、李翔昊、吕经纬、李宇杰、刘其帅、宋恒、赖景愚等同学作为该书的第一批阅读者和实践者，并对该书的内容提出了许多宝贵的修改意见。

在此，谨对关心和帮助本书完成的所有老师、同学和朋友，以及相关单位致以诚挚的感谢。

宋友 北京航空航天大学软件学院

梁士兴 IBM 中国开发中心

黄璐 IBM 中国开发中心

王鑫 IBM 中国开发中心

2009 年 4 月 北京

CONTENTS

目 录

第 1 章 导论	1
1.1 RoR 的革命与 Web 开发的新时代	1
1.2 RoR 并不完美	2
1.2.1 Ruby 语言方面的不足	2
1.2.2 对历史遗留项目的支持较为困难	2
1.3 Grails 的诞生解决了一些遗憾	3
1.3.1 Groovy 语言	3
1.3.2 Grails 站在了巨人的肩膀之上	3
1.3.3 Grails 有良好的扩展性	3
1.4 对 Grails 的一些误解	3
1.5 本书的使用说明	4
1.6 本章小结	4

第一篇 入门篇

第 2 章 Hello Grails	6
2.1 Grails 的安装	6
2.1.1 JDK 的安装与配置	6
2.1.2 Grails 的安装	7
2.2 创建 Grails 工程	8
2.3 Grails 的 MVC 架构	11
2.4 Scaffold 应用程序	14
2.5 开发工具的使用	17
2.6 本章小结	19
第 3 章 Groovy VS Java	20
3.1 Groovy 的基本类型与运算符	21
3.1.1 字符串	21
3.1.2 数字	22
3.1.3 Groovy 的类	23
3.1.4 运算符	24
3.2 Groovy 的控制结构	25
3.3 Groovy 的集合	27
3.3.1 列表	27
3.3.2 映射	28
3.3.3 区间	29
3.4 Groovy 的闭包	30

3.4.1 闭包的定义	30
3.4.2 闭包的代表	31
3.4.3 闭包在 GDK 中的使用	31
3.5 本章小结	33

第二篇 实际应用

第 4 章 商品维护

4.1 准备工作	36
4.2 查看商品列表	40
4.3 创建和编辑商品	44
4.4 本章小结	48

第 5 章 商品搜索

5.1 构造查询表单	49
5.2 复杂的数据库查询	50
5.2.1 HibernateCriteriaBuilder 的初窥	51
5.2.2 数据库的分页查询	54
5.2.3 将查询改造为 inner join	59
5.3 显示分页导航	60
5.4 本章小结	62

第 6 章 用户注册与登录

6.1 表单验证与资源文件	63
6.2 用户注册	69
6.3 用户登录	73
6.3.1 登录的数据库查询	73
6.3.2 使用 Session 维持会话	74
6.3.3 自定义 Codec 实现 密码加密	75
6.4 登录保护	76
6.5 本章小结	79

第 7 章 购物车与订单

7.1 购物车的查看与管理	80
7.1.1 定义购物车的 Domain 类	80
7.1.2 定义 OrderService 类	82
7.1.3 显示购物车	84

7.1.4 维护购物车	85
-------------	----

7.2 订单的提交

7.2.1 定义订单的 Domain 类	90
7.2.2 提交订单的表单页面	90
7.2.3 订单的保存	94

7.3 订单的查看

7.4 本章小结	99
----------	----

第 8 章 系统后台管理

8.1 页面布局的使用	100
8.1.1 Grails Layout 的基础知识	100
8.1.2 为系统后台管理创建 统一的 decorator	103
8.2 文件上传的实现	107
8.2.1 开发表单页面	107
8.2.2 在 Controller 中接收文件	108
8.3 修改订单状态	109
8.4 本章小结	110

第 9 章 Grails 的自动化测试

9.1 Grails 自动化测试基础知识	111
9.2 编写测试用例	113
9.2.1 对 Domain 类进行测试	113
9.2.2 对 Service 类进行测试	116
9.2.3 对 Controller 进行测试	118
9.2.4 对 Taglib 进行测试	120
9.3 本章小结	121

第 10 章 部署应用

10.1 Grails 对部署的支持	122
10.2 配置应用程序	124
10.3 本章小结	127

第三篇 深入了解 Grails

第 11 章 深入 GORM

11.1 自定义映射	130
11.1.1 基本映射	130
11.1.2 配置主键	131

11.1.3 “锁”与 Version	133	14.2 GSP 标签库	175
11.1.4 事件与自动时间戳	134	14.2.1 常用的内置标签	176
11.1.5 映射 Blob 字段	134	14.2.2 开发自定义标签	179
11.1.6 定义非持久化属性	135	14.3 Grails 对 Ajax 的支持	182
11.2 深入理解 Domain 间的关系	136	14.4 本章小结	184
11.2.1 一对一关系	136		
11.2.2 一对多关系	137	第 15 章 实现 Web Service	185
11.2.3 多对多关系	139	15.1 REST 风格的 Web Service	185
11.2.4 继承关系	141	15.1.1 什么是 REST	185
11.2.5 合成关系	143	15.1.2 在 Grails 中实现 REST	185
11.3 数据库查询小结	143	15.1.3 在 Client 端调用服务	187
11.3.1 GORM 提供了便捷的 查询方法	143	15.2 基于 SOAP 的传统 Web Service	188
11.3.2 基于 HQL 的查询	145	15.3 本章小结	189
11.4 对 GORM 进行性能优化	146		
11.4.1 设置抓取模式	147	第 16 章 使用 Grails 插件	190
11.4.2 使用二级缓存	147	16.1 插件的安装	190
11.5 使用 GRAG 工具生成 Domain	151	16.2 插件的组织结构	196
11.6 本章小结	154	16.3 插件的使用	197
		16.3.1 Acegi 插件	197
第 12 章 与 Spring 整合	155	16.3.2 Debug 插件	204
12.1 依赖注入与 Spring 容器基础	155	16.4 本章小结	205
12.1.1 依赖注入	155		
12.1.2 Spring 容器基础	157	第四篇 Grails 解密	
12.2 在 Grails 中使用 Spring	158		
12.3 本章小结	160	第 17 章 高级 Groovy 特性	208
第 13 章 深入 Controller	161	17.1 动态方法调用与属性访问	208
13.1 Controller 中常用的属性与方法	161	17.1.1 动态方法调用	208
13.2 自定义 URL Mapping	164	17.1.2 动态属性访问	208
13.3 Web Flow	167	17.2 invokeMethod 和 getProperty	209
13.4 本章小结	172	17.3 MOP 动态基础	211
第 14 章 深入 Groovy Server Page	174	17.3.1 遍历方法和属性	211
14.1 GSP 基础知识	174	17.3.2 动态添加方法	213
14.1.1 GSP 输出表达式	174	17.3.3 动态添加属性	215
14.1.2 GSP 中预定义的变量 与作用域	175	17.3.4 使用方法对象	216
		17.3.5 为某一特定的实例 添加方法	217
		17.4 本章小结	218

第 18 章 Grails 插件开发	219
18.1 创建与发布插件	219
18.2 插件能做什么	221
18.2.1 添加 Spring 配置信息	223
18.2.2 与 Spring 容器交互	224
18.2.3 修改 web.xml	224
18.2.4 添加动态方法	226
18.2.5 捕获变更	227
18.3 插件的依赖关系	229
18.4 在安装或升级时执行附加操作	230
18.5 本章小结	230
第 19 章 浅析 Grails 的源程序	231
19.1 准备工作	231
19.1.1 下载源码	231
19.1.2 编译 Grails 源码	231
19.2 HibernateCriteriaBuilder 的原理	233
19.3 开启 Hibernate Query Cache	237
19.4 本章小结	241
第 20 章 未来 Grails 版本的新特性	242
20.1 GORM 的新特性	242
20.1.1 更多的 GORM 事件	242
20.1.2 映射基本类型的集合	243
20.1.3 对 Domain 的只读访问	243
20.1.4 定义默认排序字段	243
20.1.5 改进的 findBy	245
20.2 对插件系统的改进	245
20.3 数据绑定	245
20.4 在 GSP 中使用 JSP 的标签	246
20.5 加密配置文件中的数据库密码	246
20.6 本章小结	246
参考文献	247
索引	248

第1章

导论

1.1 RoR 的革命与 Web 开发的新时代

随着互联网的普及和发展,绝大多数计算机应用被设计为基于 Web 技术的 Web 应用。Web 应用有着发布快捷、使用简便、对客户端要求低等优势。通过 Web 技术,可以让企业更容易发布服务,也可以让用户更容易享受信息时代的便捷。Web 的优势导致了 Web 开发的需求呈现出爆炸式的增长,进一步则对 Web 开发技术提出了更高的要求。

RoR (Ruby on Rails) 的诞生就创造了一个这样的奇迹,它在短短的几年间,在无数如 JSP、PHP 等元老级的 Web 技术面前脱颖而出,并且给了整个 J2EE 世界以极大的震撼!正如 RoR 所标榜的,它的开发效率是 J2EE 的 5 倍以上。这样巨大的差异不可能不引起 Java 世界的反思。

首先,为什么 Rails 这样成功的开发框架会诞生于 Ruby 语言,而且在其出现之后,Java 世界并没有产生能与之抗衡或者模仿它的框架呢?其次,Java 在企业级开发中一直追求的重量级的大而全,真的是用户的最想要吗?

带着这样的几个问题,这里不妨简单分析一下 RoR 所取得的成功。

首先 RoR 的优雅与快捷,相当程度上源于动态脚本语言 Ruby 的“元编程”技术。

所谓动态语言,一般有两层含义:动态类型和动态结构。RoR 的基础——Ruby 语言,同时符合上面的两种特性。支持动态类型,指只在运行时确定 Ruby 的变量类型信息;支持动态结构,指利用“元编程”技术,在程序运行时,对 Ruby 类的属性和方法进行动态增减。“元编程”的技术可以用于实现更灵活的 Web 框架,它是 Java 等静态语言所不具备的特性,因而很遗憾,Java 不可能实现出 Rails 这样的框架。

谈概念,总是让人昏昏欲睡。不妨看看例子。假设数据库的 Goods 表包含 name 和 title 字段,那么针对这两个字段进行联合查询的代码如下:

```
def goodsList = Goods.findAllByNameAndTitle(name, title)
```

用传统方法(如 Java)设计框架,想实现提供类似 findAllByNameAndTitle 这样的方法,是不现实的。因为仅 name 和 title 就会有“name and title”和“name or title”两种条件组合(考虑到 name 和 title 前后位置替换,虽然逻辑相同,但对应的函数名不同,所以应该是排列),再加上其他属性,产生的查询和排序会呈现出爆炸式的增长,再聪明的框架也不可能预先准备好应付所有的情况。通常需要用户自己去实现相关的查询逻辑,即自己

编程实现 `findAllByNameAndTitle`。显然，利用 Ruby 的“元编程”技术动态添加方法，在框架级别实现支持这样的动态查询，无疑会使开发变得简单很多。而 Java 这样的静态语言则很难实现这样的功能。因此，可以这样理解，Rails 利用了 Java 不支持的动态脚本语言的特性，实现了许多更有利于减轻用户开发劳动的功能，从而让用户觉得 Rails 更简单快捷。Rails 正是因为这样而取得了成功。

RoR 的设计指导思想就是简化开发过程。传统的 Java 开发方式，其设计显得过于学术化，而忽略了用户的感受。典型的如 SSH（Struts + Spring + Hibernate）的开发方式，为了追求灵活性，系统的各层都可以替换成其他的框架。虽然灵活，但对简化开发带来的好处却非常有限。RoR 则设计得非常实用，提供一个一栈式的解决方案，各层的设计都只有一个目标，那就是简单！

虽然 RoR 比 J2EE 简单，但 RoR 同样有着良好的组织结构。PHP 也是十分简单易用的 Web 开发技术，与 PHP 相比，Rails 最大的优势就在于它的组织结构更加清晰。RoR 是一个 MVC（Model-View-Control，模型-视图-控制器）的 Web 框架，它在工程中预先定义了若干的文件夹，分别用于存放模型-视图-控制器，其结构非常清晰明了。这体现 RoR 的另一个理念，即约定优于配置：约定不同的文件夹中存放特定类型的文件，而无需进行配置。

1.2 RoR 并不完美

RoR 凭借着它极高的开发效率和飞速增长的用户数量，毫无疑问地成为了现如今最成功的 Web 开发技术之一。然而，RoR 并不完美，它在几个方面还存在着一定的不足。

1.2.1 Ruby 语言方面的不足

虽然 Ruby 也是一种优秀的脚本语言，但它存在着执行性能差、GC（Garbage Collection，垃圾回收）缺陷等不足。

Ruby 的语法相对 Java 有较大的区别，对 Java 的程序员而言，需要学习一门全新的编程语言，对编程习惯有较大的冲击。

基于 Ruby 的开源项目较少，一些常见的开发过程中的问题，如 workflow、全文检索等，都还缺乏成熟的解决方案。

1.2.2 对历史遗留项目的支持较为困难

Java 仍然是现在最流行的编程语言，在企业级别，有大量基于 J2EE 技术的历史遗留项目。对于这样的企业，如果采用 Ruby on Rails 开发新系统，则意味着难以实现对历史遗留项目的支持。

1.3 Grails 的诞生解决了一些遗憾

Grails 的全称为 Groovy on Rails，是一种基于 Groovy 语言的 Rails 类框架。它继承了 RoR 的绝大多数优点：Grails 几乎和 RoR 一样简单，甚至在某些方面，比 RoR 更简单强大（如数据库的查询访问）；Grails 同样也遵循约定优于配置的指导思想。除此之外，它还有如下几个不同于 RoR 的特点。

1.3.1 Groovy 语言

Groovy 是与 Ruby 不同的脚本语言，虽然 Groovy 也是脚本语言，但它是以 Java 字节码的方式运行在 Java 虚拟机之上的。这样有两方面好处：一方面，可以有较好的性能；另一方面，Groovy 可以在语言级别透明地调用 Java 程序。因而，使用 Groovy 可以良好地兼容历史遗留项目，并且能够运行在成熟可靠的 J2EE Server 之上。

Groovy 有着介于 Java 和 Python 之间的语法风格，灵活不失稳重。给 Java 程序员降低了学习的门槛。

1.3.2 Grails 站在了巨人的肩膀之上

Grails 在实现过程中，使用了大量成熟可靠并为世人所广为认可的 Java 开源项目。使用了 Spring MVC 框架作为底层，实现对 Web MVC 的支持；使用了 Hibernate 作为底层，实现 GORM（GORM 是 Grails 提供的对象关系映射框架）；使用 Spring IoC 容器，实现对系统各个模块的组织与管理；使用 SiteMesh，实现对页面布局（layout）的统一和简化。

有了这些成熟可靠的项目作为基础，Grails 的可靠性得到了保证，复杂性得到了降低。从而使得 Grails 可以在更高的起点上，更有效地解决实际开发过程中的问题。

1.3.3 Grails 有良好的扩展性

Grails 采用了插件化的设计架构，不仅现有的功能是通过插件实现的（系统插件），还可以通过插件的方式，加入更多新功能，或者对其他的一些流行框架（如 Spring Security、jQuery 等）进行整合。

目前 Grails 官方网站上，提供了近百个插件，用以解决方方面面的问题。有效地利用这些插件，可以显著地提高开发效率和降低开发的难度。

1.4 对 Grails 的一些误解

作为新兴事物的 Grails，人们还对其存在一些误解^[1]。

(1) Grails 还不够成熟。

事实上,已经有越来越多的组织和个人选用 Grails 技术开发网站,Grails 官方站点已经列出了上百个使用 Grails 技术的成功商业案例^[2]。此外,由于 Grails 本身是基于 Hibernate、Spring 和 SiteMesh 这些成熟完善的框架而构建的,因而无需担心它的成熟程度。

(2) Grails 是否只是 Rails 的一个克隆产物。

Ruby on Rails 引入了不少非常好的主意,Grails 将其中的一部分应用到了 Groovy/Java 世界中,并且还加入了许多 Ruby 中并不存在的特性和概念,所有这些东西都是以一种对 Groovy 和 Java 程序员有意义的方式展现出来的。

(3) 有了 JRuby on Rails 之后,谁还要 Grails 呢?

Grails 的目标与 JRuby on Rails 大为不同。它不是为了在 Groovy 语言上实现一个 Rails 的移植版本,而是将业界最为强悍的组件(如 Spring、Hibernate、Quartz、Compass 和 SiteMesh 等)以最佳方式组合起来的一个实践,并通过采纳无配置规约使它们符合“不重复(Don't Repeat Yourself, DRY)”的原则。Grails 没有重复造轮子,而且,由于 Grails 内核的绝大部分代码都是以 Java 编写的,它也显得更加强壮、稳定和高效。事实上,从内核角度看 Grails 应用,只相当于是一个 Spring MVC 应用,因而可以被部署到所有的主流容器之上,包括大型的商业容器,如 WebLogic、WebSphere 和 Oracle AS。

1.5 本书的使用说明

本书统一对代码和控制台输入输出使用 Consolas 字体,这是一种等宽的字体。具体显示效果如下:

```
def goodsList = Goods.findAllByNameAndTitle(name, title)
```

对于比较重要的内容,会标记为粗体或粗斜体,如以下代码所示:

```
def goodsList = Goods.findAllByNameAndTitle(name, title)
println goodsList
goodsList = Goods.list()
println goodsList
```

此外,对于通过控制台输入的命令,会使用“>”开头,如:

```
>cd Helloworld
```

1.6 本章小结

本章对 RoR 的成功进行了简单的分析,指出了它的一些不足之处。Grails 框架能有效地弥补 RoR 的这些不足之处,因而更适合作为 Java 程序员首选的 Web 开发技术。本章的最后对本书的字体、格式进行了简要说明。